Journal of Computational and Applied Mathematics 29 (1990) 35–50
North-Holland

A polynomial based iterative method for linear parabolic equations

Mark J. SCHAEFER *
Department of Mathematics, Texas A&M University, College Station, TX 77843, U.S.A.

Received 30 May 1988
Revised 28 February 1989

Abstract: A new polynomial based method (PBM) is developed to integrate multi-dimensional linear parabolic initial-boundary value problems. It is based on L_2-approximations to $f(z) = (1 - \exp(-z))/z$, $f(0) = 1$, over ellipses in the complex plane using expansions of f in Chebychev polynomials. The calculation of the Fourier coefficients requires numerical integration over only a single line segment in the complex plane whose recommended length and orientation depend on the step size and the parabolic operator itself. The simplicity with which these coefficients are obtained rests on special properties of the Chebychev polynomials.

Most of the work in PBM consists of matrix–vector multiplications, involving a matrix L which arises from the spatial discretization of the differential operator. To be specific, PBM integrates the semi-discrete problem $u_t = L(t)u + b(t)$, $u, b \in \mathbb{R}^n$ and $L \in \mathbb{R}^{n \times n}$, and requires only a modest amount of storage (a few vectors of order n). Due to the analyticity of f it has good convergence properties and in the numerical examples considered, it compares favorably to standard methods from the classes of Alternating Direction Implicit and Locally One-Dimensional schemes, as measured by the CPU-times required on a single CPU of a CRAY X-MP/24. It is also competitive with Crank–Nicolson which is coupled with two proven iterative solvers. I recommend PBM on problems which require high spatial accuracy and problems whose solutions contain significant high-frequency components.

Keywords: ADI/LOD methods, Chebychev polynomials, finite differences, implicit methods, iterative methods, L_0-stability, least squares approximation, parabolic equations, semi-discrete equations, sparse linear systems.

1. Introduction and overview

The purpose of this paper is to introduce a new polynomial based iterative method for approximating the solution of linear parabolic initial-boundary value problems in three space dimensions. Starting with the differential equation

$$\frac{\partial u}{\partial t} = \sum_{i,j=1}^{3} a_{ij}(x, y, z, t)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^{3} b_i(x, y, z, t)\frac{\partial u}{\partial x_i} + c(x, y, z, t)u + f(x, y, z, t), \tag{1.1}$$

and suitable initial and boundary conditions, we replace the spatial derivatives in (1.1) and any derivative boundary conditions by difference quotients and obtain a system of ODEs of the form

$$u_t = -Lu + b, \tag{1.2}$$

* Present address: Department of Mathematics and Computer Science, Virginia Military Institute, Lexington, VA 24450, U.S.A.

where u in (1.2) is a vector of nodal values approximating the solution of (1.1) at a set of mesh points. The initial condition for (1.2) is of course obtained from that for (1.1). The *semi-discrete* problem (1.2) has been discussed by many authors, including Varga [21], Fox [7] and more recently Laurie [12]. Laurie, in particular, stresses the connection between solving linear parabolic PDEs and approximating the exponential function; he assumes that L in (1.2) is diagonalizable and has real positive eigenvalues which means that the approximations can be carried out over segments of the positive real line. In parabolic problems it is true that L's spectrum, if not real, does usually lie in the right half of the complex plane. On the other hand an example in [19] shows that even in the absence of first-order derivative terms in (1.1) the matrix L can possess complex eigenvalues if one uses highly accurate difference equations which, near the boundary of the grid, are necessarily nonsymmetric. In any case we will not make any assumptions on the spectral properties of L in (1.2): in Section 2 we introduce the polynomial method (PBM) which is based on polynomial approximations in the complex plane to the entire function $(1 - e^{-z})/z$. We then show in Section 3 why the Chebychev polynomials are particularly suited to this purpose.

Among the most popular methods for solving multi-dimensional parabolic problems are the Alternating Direction Implicit (ADI) schemes and to a lesser extent the Locally One-Dimensional (LOD) schemes. In discretizing (1.2) with respect to time one often faces the difficult problem of solving very large sparse linear systems which are not easily taken care of by direct methods; ADI and LOD manage to split these equations in such a way that the resulting implicitness involves only one spatial direction at a time. The linear systems to be solved are then often tridiagonal or nearly tridiagonal and can be solved very efficiently. A drawback of these methods can be the need to generate boundary conditions for intermediate solutions, especially in nonrectangular domains. If fourth-order spatial accuracy is desired, then these schemes can become inefficient or even unstable, and we will see in Section 5 that in this situation PBM is a very competitive alternative.

Several recent papers have focused on the development of implicit finite-difference schemes for parabolic problems which are at least second-order accurate in time and in addition L_0-stable, which implies that high-frequency components in the solution are adequately damped (see [3,10,13]). A scheme which lacks this kind of stability may require rather small time steps even if it is already A-stable. Cash [3] identifies most of the novel schemes as multiple-stage Runge–Kutta methods which require the solution of a linear system at each stage. Since some of these stages share the same coefficient matrix these methods are attractive in a one- or perhaps even two-dimensional setting where direct methods of solution are feasible, but would be very expensive to implement in three space dimensions. Under fairly restrictive conditions, i.e., assuming homogeneous boundary conditions and source terms it is possible to extend these ideas in an Alternating Direction context and derive an efficient L_0-stable ADI method (see [13]). The polynomial method by construction adequately damps high-frequency components without sacrificing accuracy.

Another important class of schemes for the solution of parabolic problems are Hopscotch schemes. They are simple to implement and generally do not make any assumption on the specific form of the spatial differential operator in (1.1) as do many ADI or LOD methods. For example, Gourlay and McKee [9] report Line Hopscotch both more accurate and efficient than ADI and LOD competitors when applied to a variable-coefficient problem with a mixed derivative. On the negative side Hopscotch schemes by construction suffer from a poor local

truncation error and are only recommended if the solution sought need not be highly accurate. Because of limited space we shall not consider these schemes in this paper; the interested reader is referred to [19] where a recently proposed algorithm of this class was compared to the polynomial method on a problem with a mixed derivative and relatively low demands on the accuracy.

With the advent of powerful iterative solvers such as CHEBYCODE [2], one does not need to rely solely on splitting schemes when integrating (1.2), for fear of the extensive computational work required in solving the large and sparse linear systems that arise otherwise. Iterative methods can take advantage of the well-conditioning of the linear systems typical for small time steps and use the solution of the previous time level as an initial guess in the iteration for the next solution. Their application appears especially attractive on problems with time dependent coefficients since accuracy may then dictate small time steps, and direct methods used in conjunction with splitting schemes may become inefficient because of the need to refactor the equations with each new step. An example of this type is explored in Section 6 and we will see that for PBM the number of iterations required per step is less sensitive to the step size than it is for CHEBYCODE and CGLS [17].

A problem which may occur in the implementation of the polynomial method is the ill-conditioned computation of high-order Fourier coefficients by numerical integration. This difficulty and several remedies are addressed in Section 4.

In summary, I recommend the polynomial method for linear parabolic problems which require highly accurate solutions, e.g., fourth-order spatial accuracy, and problems whose solutions carry high-frequency components, induced for example by a rough initial condition or discontinuous boundary conditions.

2. The polynomial method

If L and b are constant and L is nonsingular the solution of (1.2) is given by

$$u(t) = e^{-Lt}u_0 + (tL)^{-1}(I - e^{-Lt})tb, \quad \text{where } u_0 = u(0). \tag{2.1}$$

The direct approximation of the exponential matrix e^{-Lt} in (2.1) is a difficult task (see [16]) and for large problems is further complicated by the fact that we cannot expect e^{-Lt} to reflect the sparsity pattern of L. On the other hand, all we need are matrix–vector products of the form $e^{-Lt}v$ for some vector v; the Chebychev rational methods advocated by Cavendish, Culham and Varga [4] achieve this by constructing vectors of the form $r(Lt)v$ for certain rational functions r. A different approach is to approximate $e^{-Lt}v$ by $p(Lt)v$ where the polynomial p is an approximation to the exponential over a suitable region in the complex plane. Saad [18] shows how to efficiently generate least squares polynomials in $\mathbb{C}$ to solve nonsymmetric matrix problems (based on knowledge of the convex hull of the spectrum) and claims that the algorithm can be extended to yield instead approximations to $e^{-Lt}v$. In any case, from (2.1) it is seen that for nonzero b we then still have to solve a system of the form $Lx = y$ which corresponds to the associated elliptic problem, and below we show how to avoid this additional work by approximating instead a slightly different matrix function.

For singular L it is necessary to consider the matrix extension of the function

$$f(z) = \begin{cases} \dfrac{1 - e^{-z}}{z}, & z \neq 0, \\ 1, & z = 0. \end{cases} \tag{2.2}$$

Note that $\lim_{z \to 0} zf(z) = \lim_{z \to 0}(1 - e^{-z}) = 0$, hence f (as defined for $z \neq 0$) has a removable singularity at 0 and with $f(0)$ defined as above is analytic for all z in $\mathbb{C}$. Given any square matrix A we may therefore define the matrix $f(A)$ via the Cauchy integral formula (see [8]):

$$f(A) = \frac{1}{2\pi i} \int_{\Gamma} f(z)(zI - A)^{-1} \, dz. \tag{2.3}$$

Here Γ is a closed contour which contains the spectrum of A in its interior. Now the solution of (1.2) can be written in the form

$$u(t) = e^{-Lt} u_0 + f(tL)tb. \tag{2.4}$$

(The validity of (2.4) is easily verified by differentiation, where (2.3) is used to show that $d/dt\, f(tL) = Lf'(tL)$; the details are omitted.) Using $e^{-Lt} = I - (tL)f(tL)$ it follows that

$$u(t) = u_0 + t f(tL)(b - Lu_0). \tag{2.5}$$

The idea now is to use a matrix polynomial $Q_n(tL)$ to approximate $f(tL)$. The computed solution $\bar{u}_n(t)$ then satisfies

$$\bar{u}_n(t) = u_0 + t\, Q_n(tL)(b - Lu_0), \tag{2.6}$$

and

$$\| u(t) - \bar{u}_n(t) \| \leqslant |t| \, \| f(tL) - Q_n(tL) \| \, \| b - Lu_0 \|, \tag{2.7}$$

for some consistent matrix norm $\| \cdot \|$. Clearly, we must be concerned with the convergence of $\| f(tL) - Q_n(tL) \|$ to zero as the degree n of Q_n tends to infinity. For L in $\mathbb{R}^{m \times m}$ this generally requires that $Q_n^{(r)}$ converge to $f^{(r)}$, $0 \leqslant r \leqslant m - 1$, over some set containing the spectrum of L. Because the approximations we will use converge uniformly to f, and uniform approximation of analytic functions by analytic functions over regions in $\mathbb{C}$ implies simultaneous uniform approximation of all derivatives, convergence of $\| f(tL) - Q_n(tL) \|$ to zero will be guaranteed.

Let us consider (2.6) from a practical point of view: $\bar{u}_n(t) = u_0 + t\, Q_n(tL)r_0$, where $r_0 = b - Lu_0$. In general we will not know in advance what degree polynomial to choose in (2.6); what is needed is an iterative procedure which generates a sequence of polynomials $\{Q_j\}$ such that $\{\bar{u}_j(t)\}$ is a sequence of successively better approximations to $u(t)$, and a criterion that tells us when to stop the iteration. Moreover, it should be feasible to progress from $\bar{u}_{n-1}(t)$ to $\bar{u}_n(t)$ with only a few vectors saved from previous iterations as these vectors may be very large. These considerations will be taken up in the next two sections.

We turn now to the more general equation

$$u_t = -L(t)u + b(t). \tag{2.8}$$

Assume $u(0) = u_0$ is known and $L(t)$ and $b(t)$ are twice continuously differentiable for $t \geqslant 0$. If we define

$$u_{\mathrm{PM}}(\Delta t) = u_0 + \Delta t\, f\big(\Delta t\, L(\tfrac{1}{2}\, \Delta t)\big)\big(b(\tfrac{1}{2}\Delta t) - L(\tfrac{1}{2}\, \Delta t)u_0\big), \tag{2.9}$$

then it is not difficult to show that $u(\Delta t) - u_{PM}(\Delta t) = O(\Delta t^3)$, i.e., u_{PM} is second-order accurate in time (see [19]). In practice it may be necessary to take many small steps in order to integrate to a fixed time T and achieve a desired accuracy. This means that the polynomial method should require fewer iterations for small steps than for large ones or it will not be competitive on problems with time-dependent coefficients. In Section 6 the Crank–Nicolson method (which is also second order in time) is coupled with two different iterative solvers and compared to the polynomial method on a problem which after spatial discretization results in (2.8) with both L and b time-dependent.

Returning once more to the case of constant L and b, we anticipate that in this situation the polynomial algorithm will not require any time stepping structure as is needed for schemes based on a temporal discretization of (1.2). This is true in principle and in particular in the example of Section 5, although more extensive experiments carried out in [19] indicate that if the time step chosen is so large (or the termination criterion so stringent) as to require an approximating polynomial of degree greater than 100, one may sometimes run into difficulties computing certain high-order Fourier coefficients needed by the algorithm. In this case it may be necessary to reduce the step size or compute the coefficients in higher precision (see Section 4).

3. Least squares approximation in the complex plane

We begin by introducing some notation taken mostly from Davis [6]: let $\{\epsilon_\rho\}$ be the family of confocal ellipses with foci at ± 1, major axis a, minor axis b, and $\rho = a + b$. Clearly $\rho \geqslant 1$, and ϵ_1 is the degenerate ellipse which coincides with the interval $[-1, 1]$. The interior of ϵ_ρ is designated by $\hat{\epsilon}_\rho$ and the closure of $\hat{\epsilon}_\rho$ by $\bar{\epsilon}_\rho$. We denote by T_n and U_n the Chebychev polynomials of degree n of the first and second kind, respectively, standardized by $T_n(1) = 1$ and $U_n(1) = n + 1$.

Theorem 1. *Let g be analytic in an open region R of the complex plane containing the real line segment $[-1, 1]$. Let*

$$p_n(z) = \sum_{i=0}^{n} a_i T_i(z), \tag{3.1}$$

and

$$q_n(z) = \sum_{i=0}^{n} b_i U_i(z), \tag{3.2}$$

with

$$a_0 = \frac{1}{\pi}\int_{-1}^{1} (1-x^2)^{-1/2} g(x) T_0(x)\, dx,$$

$$a_i = \frac{2}{\pi}\int_{-1}^{1} (1-x^2)^{-1/2} g(x) T_i(x)\, dx, \quad i > 0,$$

$$b_i = \frac{2}{\pi}\int_{-1}^{1} (1-x^2)^{1/2} g(x) U_i(x)\, dx.$$

The sequences $\{p_j^{(m)}\}$, $\{q_j^{(m)}\}$, $m \geqslant 0$, converge uniformly to $g^{(m)}$ in every ellipse $\bar{\epsilon}_\rho$ contained in R. Furthermore, if g is entire $\lim_{i\to\infty} |a_i|^{1/i} = \lim_{i\to\infty} |b_i|^{1/i} = 0$.

Proof. See Szegö [20]. □

The fact that analytic functions can be expanded in series of Chebychev polynomials is of course well known and follows directly from the completeness property of $\{T_j\}$ and $\{U_j\}$ in certain complex inner product spaces (see [6]). We remark in passing that since uniform convergence implies L_2-convergence, Theorem 1 implies that the expansions p_n and q_n are optimal among all polynomials of degree n as measured in the inner product induced norms of these spaces. It is remarkable however that the approximations p_n and q_n can be generated with coefficients defined only in terms of *real* inner products. Note that if $Q_n(tL)$ in (2.6) is obtained via (3.1) or (3.2) with z replaced by tL and g by f as in (2.2), then, since f is entire, $Q_n(tL)$ will always converge in norm to $f(tL)$ as n tends to infinity, for any square matrix L. In fact we have the following result.

Theorem 2. *Suppose* $Q_n(tL)$ *in* (2.6) *is obtained via the expansions* (3.1) *or* (3.2). *Then* $\lim_{n\to\infty} \| u(t) - \bar{u}_n(t) \|_2^{1/n} = 0$, *where* $u(t)$ *is as in* (2.5).

Proof. See [19]. □

The vectors $\{Q_j(tL)r_0\}$ which are required in (2.6) are conveniently generated by taking advantage of the three-term recurrence satisfied by $\{T_j\}$ and $\{U_j\}$. Using for example (3.2) we have

$$Q_n(tL)r_0 = \left(\sum_{i=0}^{n} b_i U_i(tL) \right) r_0 = b_n U_n(tL)r_0 + Q_{n-1}(tL)r_0,$$

and, for $n \geqslant 2$,

$$U_n(tL)r_0 = 2tL(U_{n-1}(tL)r_0) - U_{n-2}(tL)r_0. \tag{3.3}$$

Hence

$$\bar{u}_n(t) = \bar{u}_{n-1}(t) + tb_n U_n(tL)r_0, \tag{3.4}$$

where b_n is obtained by numerical integration and $U_n(tL)r_0$ from (3.3) which involves essentially one matrix–vector multiplication. Only three vectors from previous steps need be saved: $\bar{u}_{n-1}(t)$, $U_{n-1}(tL)r_0$, and $U_{n-2}(tL)r_0$.

Before concluding this section it deserves mention that [6, Theorem 12.4.7] is a theorem not unlike the theorem above, which is based on the Legendre rather than the Chebychev polynomials. In particular, iteration (3.4) could also be carried out using Legendre polynomials although this has not been realized in either of the examples below. One reason for preferring Chebychev polynomials is related to the computation of the Fourier coefficients $\{a_j\}$ and $\{b_j\}$: after a simple change of variable the corresponding integrals are of the form $\int_0^\pi h(x) \cos jx \, dx$ and $\int_0^\pi h(x) \sin x \sin(j+1)x \, dx$ with $h(x) \equiv f(\cos x)$. Integrals of this type are not uncommon and there exist special integration routines designed to handle them efficiently; in the numerical examples of Sections 5 and 6 subroutine D01ANF from the NAG FORTRAN Library (Mark 11, November 83) is used for this purpose.

4. Practical considerations

Contrary to the situation of solving linear systems, iteration (3.4) does not produce diminishing residuals, and thus residual norms are not available as an indicator for when to stop the iteration. We can, however, monitor the closeness of successive iterates and base the termination of (3.4) on this measure, a strategy which has worked well in practice (see Sections 5 and 6).

We turn now to the choice of foci (A, B) on which to base the actual computations. Given the matrix L and a stepsize Δt the choice $A = -1$, $B = +1$, which has been tacitly assumed up until now, may not be best for reasons explained below. Let $F(A, B)$ denote the family of confocal ellipses with foci at points A and B in $\mathbb{C}$. The Chebychev expansions of f analogous to (3.1) and (3.2) but with the basis polynomials translated to be orthogonal over the line segment joining A and B will be referred to as *corresponding to* $F(A, B)$. In the following a simple and heuristic strategy is presented for selecting a set of foci (A, B). To begin with we assume that we are given a rectangular region R whose purpose it is to give us an idea of the general location of $(\Delta t\, L)$'s spectrum. Since L is assumed real we may take R to be symmetric with respect to the real axis. Such a rectangle might be obtained by estimating the extreme eigenvalues of the symmetric and skew-symmetric parts of L (see [11]). In parabolic problems the symmetric part of L often stems from diffusion terms in the original PDE and is then positive definite. Even if it is perturbed and not definite we may still estimate the algebraically smallest eigenvalue at zero if it is small compared to the largest eigenvalue, and use Gershgorin's Theorem to estimate the latter. Given R we then take as foci the midpoints of the two shorter sides of R. Note that if R's horizontal side exceeds its vertical side, then the original interval $[-1, 1]$ is simply translated and scaled along the real axis, but if R is vertical the interval is also rotated. It is of some interest to note that in this case the algorithm can still be carried out in real arithmetic. We illustrate this for foci at $\pm \mathrm{i}$, using Chebychev polynomials of the first kind as the orthogonal basis over $[-\mathrm{i}, \mathrm{i}]$:

$$T_0(\Delta t\, L)r_0 = r_0, \qquad T_1(\Delta t\, L)r_0 = -\mathrm{i}\Delta t\, Lr_0,$$
$$T_n(\Delta t\, L)r_0 = -2\mathrm{i}\Delta t\, L(T_{n-1}(\Delta t L)r_0) - T_{n-2}(\Delta t\, L)r_0, \quad n \geqslant 2.$$

Hence, for n even, $T_n(\Delta t\, L)r_0$ is real and otherwise pure imaginary. Consider now the Fourier coefficients $\{a_j\}$:

$$a_0 = \frac{1}{\pi}\int_{-1}^{1}(1-x^2)^{-1/2}f(\mathrm{i}x)\,\mathrm{d}x, \qquad a_n = \frac{2}{\pi}\int_{-1}^{1}(1-x^2)^{-1/2}f(\mathrm{i}x)T_n(x)\,\mathrm{d}x, \quad n \geqslant 1.$$

Writing $f(z) = f_{\mathrm{R}}(z) + \mathrm{i}f_{\mathrm{I}}(z)$ with f_{R} and f_{I} both real-valued functions of z, it follows from (2.2) that over $[-\mathrm{i}, \mathrm{i}]$ f_{R} is even and f_{I} is odd. Since T_n is even for n even and odd for n odd, a_n is real for n even and imaginary for n odd. The quantity we need, however, is $a_nT_n(\Delta t\, L)r_0$ (cf. iteration (3.4)) and from what has just been said it follows that this vector is real for all n.

Table 1 summarizes the formulas needed to carry out the iteration (3.4) (or the corresponding one using Chebychev polynomials of the first kind), both for the case of real- and complex-valued foci. Numerical calculations such as those in Sections 5 and 6 have shown little difference between using either kind of polynomials, except that the computation of high-order Fourier coefficients appears slightly better conditioned for the second kind.

This strategy of placing foci achieves a balance between the distance d between foci and the overall size of the spectrum: if d were chosen much larger, the convergence of the iteration would be slowed considerably because even the smallest ellipse in $F(A, B)$ which contains the

Table 1

Chebychev polynomials of the first kind	Chebychev polynomials of the second kind
Real-valued foci at a and b, $a < b$; $t = \frac{1}{2}[(b-a)\cos(s) + b + a]$; $r_0 = b - Lu_0$	
$T_0(tL)r_0 = r_0$	$U_0(tL)r_0 = r_0$
$T_1(tL)r_0 = \frac{1}{b-a}(2tL - (b+a)I)r_0$	$U_1(tL)r_0 = \frac{2}{b-a}(2tL - (b+a)I)r_0$
$T_n(tL)r_0 = \frac{2}{b-a}(2tL - (b+a)I)T_{n-1}(tL)r_0 - T_{n-2}(tL)r_0,\ n \geqslant 2$	$U_n(tL)r_0 = \frac{2}{b-a}(2tL - (b+a)I)U_{n-1}(tL)r_0 - U_{n-2}(tL)r_0,\ n \geqslant 2$
$a_0 = \frac{1}{\pi}\int \frac{1-\exp(-t)}{t}\,\mathrm{d}s$	$b_n = \frac{2}{\pi}\int \frac{1-\exp(-t)}{t}\sin(s)\sin(ns+s)\,\mathrm{d}s$
$a_n = \frac{2}{\pi}\int \frac{1-\exp(-t)}{t}\cos(ns)\,\mathrm{d}s,\ n > 0$	
Complex-valued foci at $a+\mathrm{i}b$ and $a-\mathrm{i}b$, a real, $b > 0$; $t = a + \mathrm{i}bx$; $x = \cos(s)$; $r_0 = b - Lu_0$	
$f_R(x) = \mathrm{REAL}\left[\frac{1-\exp(-t)}{t}\right] = \frac{\exp(-a)(bx\sin(bx) - a\cos(bx)) + a}{a^2 + b^2x^2}$	
$f_I(x) = \mathrm{IMAG}\left[\frac{1-\exp(-t)}{t}\right] = \frac{\exp(-a)(a\sin(bx) + bx\cos(bx)) - bx}{a^2 + b^2x^2}$	
$T_0(tL)r_0 = r_0$	$U_0(tL)r_0 = r_0$
$T_1(tL)r_0 = \frac{\mathrm{i}}{b}(aI - tL)r_0$	$U_1(tL)r_0 = \frac{2\mathrm{i}}{b}(aI - tL)r_0$
$T_n(tL)r_0 = \frac{2\mathrm{i}}{b}(aI - tL)T_{n-1}(tL)r_0 - T_{n-2}(tL)r_0,\ n \geqslant 2$	$U_n(tL)r_0 = \frac{2\mathrm{i}}{b}(aI - tL)U_{n-1}(tL)r_0 - U_{n-2}(tL)r_0,\ n \geqslant 2$
$a_0 = \frac{1}{\pi}\int f_R(x)\,\mathrm{d}s$	$b_n = \frac{2\mathrm{i}}{\pi}\int f_I(x)\sin(s)\sin(ns+s)\,\mathrm{d}s,\ n$ odd
$a_n = \frac{2\mathrm{i}}{\pi}\int f_I(x)\cos(ns)\,\mathrm{d}s,\ n$ odd	$b_n = \frac{2}{\pi}\int f_R(x)\sin(s)\sin(ns+s)\,\mathrm{d}s,\ n$ even
$a_n = \frac{2}{\pi}\int f_R(x)\cos(ns)\,\mathrm{d}s,\ n$ even, $n > 0$	

All integrals in this table are taken from 0 to π.

spectrum of $\Delta t\, L$ would cover an unnecessarily large region in $\mathbb{C}$. (The expansions are optimal over ellipses whose foci are at A and B.) On the other hand, it must be warned that d should never be chosen smaller than the maximum of the two sides of R. Due to cancellation errors, the computation of the Fourier coefficients becomes difficult or impossible as the value of the integral approaches the machine epsilon or becomes less than it, and if d is chosen too small, this can become a problem because the Fourier coefficients a_k or b_k will then approach zero rapidly while $\| U_k(tL)r_0 \|_2$ or $\| T_k(tL)r_0 \|_2$ will increase rapidly. When this particular difficulty is encountered one needs to resort to one or more of the following strategies:

(1) reduce the step size;
(2) relax the termination criterion;
(3) re-evaluate the placement of foci;
(4) compute the Fourier coefficients in higher precision.

According to the heuristic placement of foci explained above, the distance between foci will be proportional to the step size Δt and thus a decrease in Δt would seem to result in a reduction rather than increase of the number of Fourier coefficients that can be computed in a given

precision. However, the primary effect of a smaller Δt is to require a lower degree polynomial and hence the computation of fewer coefficients to begin with.

Within its range of applications I regard this source of difficulty as the only drawback to the polynomial method. It has not occurred frequently and should not be considered a major obstacle to the implementation of the method.

5. The heat equation

The problem we wish to solve is the constant coefficient heat equation in the unit cube:

$$u_t = u_{xx} + u_{yy} + u_{zz}, \quad 0 \leqslant x, y, z \leqslant 1, \quad t > 0. \tag{5.1}$$

The initial and boundary conditions are given by

$$u(x, y, z, 0) = g(x, y, z) + \sum_{k=1}^{n} \frac{1}{k} \sin\left(k + \tfrac{1}{2}\right)\pi x \sin\left(k + \tfrac{1}{2}\right)\pi y \cos\left(k + \tfrac{1}{2}\right)\pi z, \tag{5.2}$$

$$g(x, y, z) = \mathrm{e}^{-\pi x/\sqrt{2}} \cos\left(\tfrac{1}{2}\pi y\right) \sin\left(\tfrac{1}{2}\pi z\right),$$

and $u = g$ over the faces $x = 0$, $y = 0$, $z = 1$, $u_x = g_x$ over $x = 1$, $u_y = g_y$ over $y = 1$, and $u_z = g_z$ over $z = 0$. The exact solution is

$$u(x, y, z, t) = g(x, y, z) + \sum_{k=1}^{n} \frac{1}{k} \mathrm{e}^{-3(k+1/2)^2\pi^2 t} \sin\left(k + \tfrac{1}{2}\right)\pi x \sin\left(k + \tfrac{1}{2}\right)\pi y \cos\left(k + \tfrac{1}{2}\right)\pi z. \tag{5.3}$$

The objective is to compute an accurate approximation to u at $t = T = 0.01$. Note that this is not a small time interval for our problem since even the lowest-order transient mode ($k = 1$ in (5.3)) is reduced in amplitude by 49% during this time period.

An ADI method (originally proposed by Jim Douglas) for the solution of (5.1) is given by

$$\begin{aligned}
\left(1 - \alpha\delta_x^2\right)U_{ijk}^{n+1*} &= \left[\frac{\alpha}{\beta}\left(1 + \beta\delta_x^2\right) + (\alpha + \beta)\left(\delta_y^2 + \delta_z^2\right)\right]U_{ijk}^{n}, \\
\left(1 - \alpha\delta_y^2\right)U_{ijk}^{n+1**} &= U_{ijk}^{n+1*} - \beta\delta_y^2 U_{ijk}^{n}, \\
\left(1 - \alpha\delta_z^2\right)U_{ijk}^{n+1} &= \frac{\beta}{\alpha} U_{ijk}^{n+1**} - \frac{\beta^2}{\alpha}\delta_z^2 U_{ijk}^{n},
\end{aligned} \tag{5.4}$$

where U_{ijk}^{n} is an approximation to $u(ih, jh, kh, n\,\Delta t)$, h is the uniform mesh spacing, $\delta_x^2 U_{ijk}^{n} = U_{i+1,jk}^{n} - 2U_{ijk}^{n} + U_{i-1,jk}^{n}$, and similar definitions apply to $\delta_y^2 U_{ijk}^{n}$ and $\delta_z^2 U_{ijk}^{n}$. With $\alpha = \frac{1}{2}(\Delta t/h^2 - \frac{1}{6})$, $\beta = \frac{1}{2}(\Delta t/h^2 + \frac{1}{6})$, scheme (5.4) is second order in time and locally fourth order in space. The boundary conditions for the intermediate solutions U^* and U^{**} are chosen consistent with (5.4) to prevent a loss of accuracy (see [15]). In order to preserve global fourth-order spatial accuracy we need at least third-order formulas to approximate the Neumann boundary conditions (see [1]), and fourth-order formulas may still be better for a given value of h. We will see shortly that there are certain pitfalls which can arise from the implementation of

such high-order difference equations at boundary grid points, which are necessarily nonsymmetric.

The following LOD scheme is also second order in time and locally fourth order in space:

$$\begin{aligned}
\left(1-\alpha\delta_x^2\right)U_{ijk}^{n+1^*} &= \left(1+\beta\delta_x^2\right)U_{ijk}^{n},\\
\left(1-\alpha\delta_y^2\right)U_{ijk}^{n+1^{**}} &= \left(1+\beta\delta_y^2\right)U_{ijk}^{n+1^*},\\
\left(1-\alpha\delta_z^2\right)U_{ijk}^{n+1} &= \left(1+\beta\delta_z^2\right)U_{ijk}^{n+1^{**}},
\end{aligned} \tag{5.5}$$

where as before $\alpha = \frac{1}{2}(\Delta t/h^2 - \frac{1}{6})$ and $\beta = \frac{1}{2}(\Delta t/h^2 + \frac{1}{6})$. Traditionally, U^{n+1^*} and $U^{n+1^{**}}$ have been thought of as approximations to $u((n+\frac{1}{3})\,\Delta t)$ and $u((n+\frac{2}{3})\,\Delta t)$, respectively, and the boundary conditions for these intermediate solutions have been suggested in accordance with

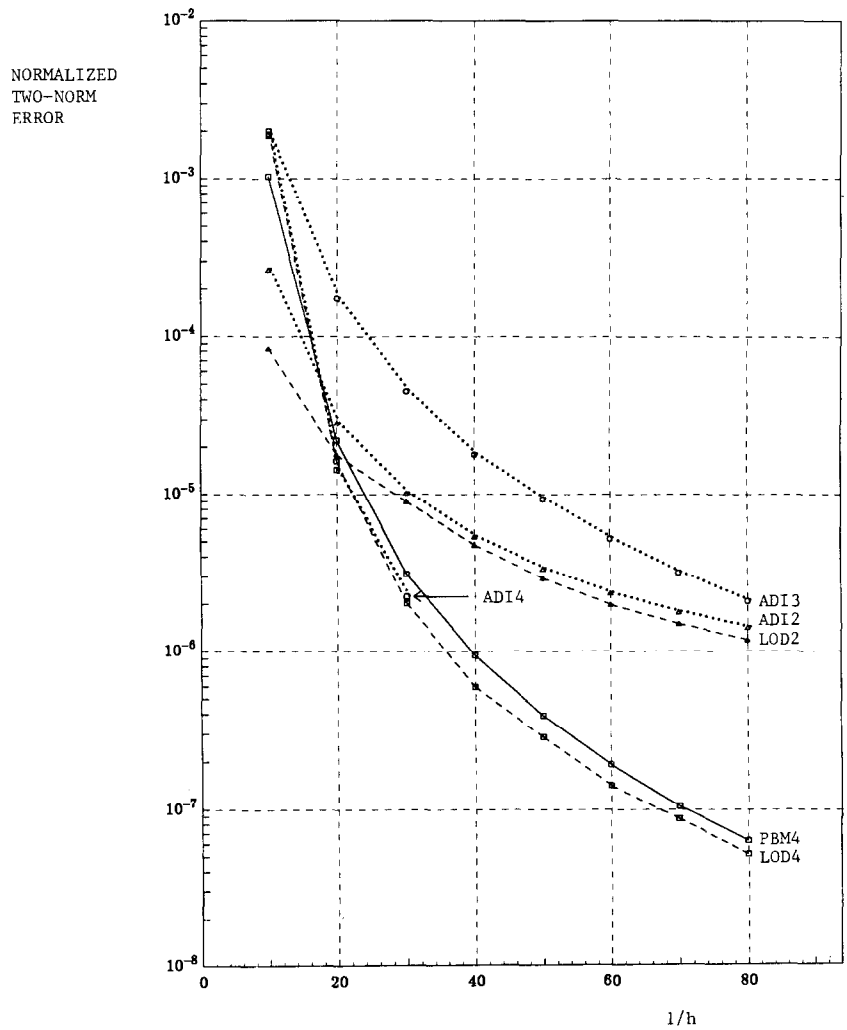

Fig. 1. Accuracy versus spatial discretization for several schemes applied to the heat equation (5.1).

Table 2
CPU-times

h	ADI2	ADI3	ADI4	LOD2	LOD4	PBM4
1/10	0.4	0.2	0.7	0.1	0.1	0.1
1/20	1.9	0.9	9.3	1.4	1.4	0.6
1/30	5.2	2.6	52.4	4.1	8.2	1.9
1/40	11.5	5.8	–	13.4	26.9	6.9
1/50	21.0	10.5	–	32.5	65.0	9.5
1/60	34.8	34.8	–	68.4	150.4	20.4
1/70	53.7	53.7	–	116.7	297.0	31.1
1/80	80.1	80.1	–	187.9	563.6	87.1

this interpretation. It was observed, however, that this choice of intermediate boundary conditions causes a loss of accuracy and in 1985 LeVeque [14] showed how to derive boundary conditions for U^{n+1^*} and $U^{n+1^{**}}$ such that no accuracy is lost. These improved boundary conditions were used in the implementation of (5.5) below. With regard to the order of approximation of the Neumann boundary conditions, the remarks made above apply here as well.

Figure 1 shows the results of applying schemes (5.4), (5.5), and the polynomial method to the model problem with n in (5.2) equal to 10 and $h = 1/10,\ 1/20, \ldots, 1/80$. The number of gridpoints (unknowns) is $N = (1/h)^3$. For each method and each value of h is plotted the two-norm error of U, i.e., $(\Sigma_{i,j,k}(U_{ijk} - u(ih,\ jh,\ kh))^2/N)^{1/2}$, at time $t = T = 0.01$. Table 2 presents the corresponding CPU-times (in seconds) obtained for the various schemes and values of h on one CPU of a CRAY X-MP/24.

Figure 1 contains the plots of three ADI schemes: ADI2 refers to (5.4) in conjunction with a second-order approximation to the Neumann boundary conditions; ADI3 stands for (5.4) using a third-order approximation; and ADI4 combines (5.4) with a fourth-order formula. Since the time steps taken are uniform, the (nearly) tridiagonal linear systems are factored only once and then repeatedly backsolved. For a given value of h, the stepsize Δt is roughly determined as being the largest for which a further reduction in Δt no longer results in a significant reduction of the error, which means that the error in the solution is dominated by the spatial truncation error. Table 3 shows the stepsizes used for the different schemes. It is interesting to observe that among these three schemes the most useful one (over the range of values of h considered) is

Table 3

h	ADI2	ADI3	ADI4	LOD2	LOD4	PBM4
	Number of steps $T/\Delta t$					Number of iterations
1/10	20	10	40	5	5	27
1/20	20	10	100	20	20	37
1/30	20	10	200	20	40	53
1/40	20	10	–	30	60	70
1/50	20	10	–	40	80	86
1/60	20	20	–	50	110	103
1/70	20	20	–	55	140	119
1/80	20	20	–	60	180	135

ADI2 even though it is only second order in space. ADI3 is third order in space but has a larger error throughout and ADI4 is almost completely useless because it is *unstable*: very small values of Δt are required to achieve stability and accuracies similar to LOD4 and PBM4 (to be discussed below), making the scheme extremely inefficient.

Two LOD methods are implemented: LOD2 and LOD4, defined similarly to ADI2 and ADI4, but of course based on (5.5) instead of (5.4). The former is a viable though not particularly efficient scheme; the problem with the latter is that in order to effect an error which behaves as $O(h^4)$ the stepsize Δt has to be reduced significantly as h decreases.

Finally, Fig. 1 also shows the graph of PBM4, the polynomial method based on fourth-order difference equations at all grid points and fourth-order difference replacements of the Neumann boundary conditions. A single step is taken from $t = 0$ to $t = T$ and the iteration stopped as soon as $\| \bar{u}_{k+1} - \bar{u}_k \|_2 / \| \bar{u}_{k+1} \|_2 \leqslant 10^{-8}$, for all values of h. The iteration is based on the Chebychev polynomials of the second kind but those of the first kind yield almost identical results. A good upper bound for $(\Delta t\, L)$'s largest eigenvalue is $20\ \Delta t/h^2$; accordingly, the foci are placed at (0, 0) and $(20\ \Delta t/h^2, 0)$. If instead we use Gershgorin's Theorem to estimate $(\Delta t\, L)$'s largest eigenvalue, the result is about $30\ \Delta t/h^2$. Using this value for one of the foci and again zero for the other the number of iterations increase by about 20% for all values of h while the accuracies of the solutions remain the same. From Table 2 and Fig. 1 it is seen that PBM4 is about as expensive though more accurate that ADI2, and about as accurate but much more efficient than LOD4. (Table 3 shows the number of iterations obtained for all values of h.)

The ADI and LOD schemes (5.4) and (5.5) derive their second-order temporal accuracy from the fact that they are ultimately based on the Crank–Nicolson scheme. It is well known that Crank–Nicolson does not adequately damp high-order modes (it is not L_0-stable). We now change the value of n in (5.2) from 10 to 40 and observe the effects on PBM4 and ADI2, leaving everything else unchanged. The results are tabulated in Table 4 where $h = 1/80$. As might be anticipated, ADI2 is much more sensitive to this change than PBM4. The polynomial method is particularly attractive for problems which contain significant high-frequency components.

6. A problem with dominant first-order derivatives

The problem we wish to solve in this section is similar to one treated by Ciment et al. [5] except for the presence of a third space dimension:

$$\begin{aligned} u_t = {} & 0.0001 \frac{x + 0.1}{(y + 0.4)(z + 0.2)(t + 1)^2} u_{xx} + 0.9998 \frac{x + 0.1}{t + 1} u_x \\ & + 0.0001 \frac{y + 0.4}{(x + 0.1)(z + 0.2)(t + 1)^2} u_{yy} + \frac{y + 0.4}{t + 1} u_y \\ & + 0.0001 \frac{z + 0.2}{(x + 0.1)(y + 0.4)(t + 1)^2} u_{zz} - 1.0001 \frac{z + 0.2}{t + 1} u_z, \\ & 0 \leqslant x,\ y,\ z \leqslant 1,\ t \geqslant 0, \end{aligned} \tag{6.1}$$

with Dirichlet boundary conditions and an initial condition at $t = 0$ consistent with the solution $u(x, y, z, t) = e^{(x+0.1)(y+0.4)(z+0.2)(t+1)}$. The objective is to find an approximate solution at $T = 1.0$ using once again fourth-order difference equations.

Table 4

n	Method	Number of iterations	Number of time steps	Two-norm error
10	PBM4	135	1	$0.627 \cdot 10^{-7}$
40	PBM4	138	1	$0.634 \cdot 10^{-7}$
10	ADI2	–	20	$0.142 \cdot 10^{-5}$
40	ADI2	–	20	$0.196 \cdot 10^{-2}$
40	ADI2	–	30	$0.905 \cdot 10^{-5}$

Let L be the matrix which results from the discretization of the right-hand side of (6.1) and (with an obvious notation) let $L = L_x + L_y + L_z$. An ADI scheme for the solution of (6.1) is given by:

$$\begin{aligned}
&\left(I - \tfrac{1}{2}\,\Delta t\, L_x^{n+1/2}\right)U^{n+1*} = \left(I + \tfrac{1}{2}\,\Delta t\, L_x^{n+1/2} + \Delta t\left(L_y^{n+1/2} + L_z^{n+1/2}\right)\right)U^n,\\
&\left(I - \tfrac{1}{2}\,\Delta t\, L_y^{n+1/2}\right)U^{n+1**} = U^{n+1*} - \tfrac{1}{2}\,\Delta t\, L_y^{n+1/2}U^n,\\
&\left(I - \tfrac{1}{2}\,\Delta t\, L_z^{n+1/2}\right)U^{n+1} = U^{n+1**} - \tfrac{1}{2}\,\Delta t\, L_z^{n+1/2}U^n.
\end{aligned} \tag{6.2}$$

Here again U^* and U^{**} are intermediate solutions which assume special boundary values consistent with (6.2) (see [15]). Scheme (6.2) is reminiscent of (5.4) which was used to solve the constant coefficient heat equation, but differs from the latter in requiring the solution of pentadiagonal rather than tridiagonal systems of equations in order to obtain fourth-order spatial accuracy. Actually (5.4) can be modified to accomodate variable coefficients but the derivation proceeds on the assumption that no first-order derivatives are present; for details, see [15] and the discussion in [1]. In [5] it is shown how to retain tridiagonality even in the presence of mixed-order derivatives; however, the approach (called the Operator Compact Implicit Method) already becomes rather complicated for two-dimensional problems when applied in an Alternating Direction fashion because of the need to generate boundary values for intermediate solutions according to a special procedure. The extension of this method to three dimensional problems is not discussed.

The fact that the systems of equations in (6.2) are functions of time and need to be refactored at each time step makes (6.2) an inefficient scheme for small Δt. Moreover, in using direct solvers we are not taking advantage of the property that for small Δt the linear systems in (6.2) may be very well conditioned. In this situation iterative methods are clearly attractive; when they are used, however, there appears no advantage in an ADI formulation over the unfactored Crank–Nicolson scheme since we do not have to fear any fill-in often associated with direct methods, and so we integrate the latter:

$$\left(I - \tfrac{1}{2}\,\Delta t\, L^{n+1}\right)U^{n+1} = \left(I + \tfrac{1}{2}\,\Delta t\, L^n\right)U^n. \tag{6.3}$$

As initial guess for the iterative solution of (6.3) we pass the solution from the previous time step which in parabolic problems is a natural first approximation to the solution at the next time level. The two iterative methods used are CHEBYCODE [2], an adaptive routine based on the Chebychev polynomials which assumes that the eigenvalues of $(I - \frac{1}{2}\,\Delta t\, L)$ are in the right half of the complex plane, and CGLS [17] which is mathematically equivalent to the method of

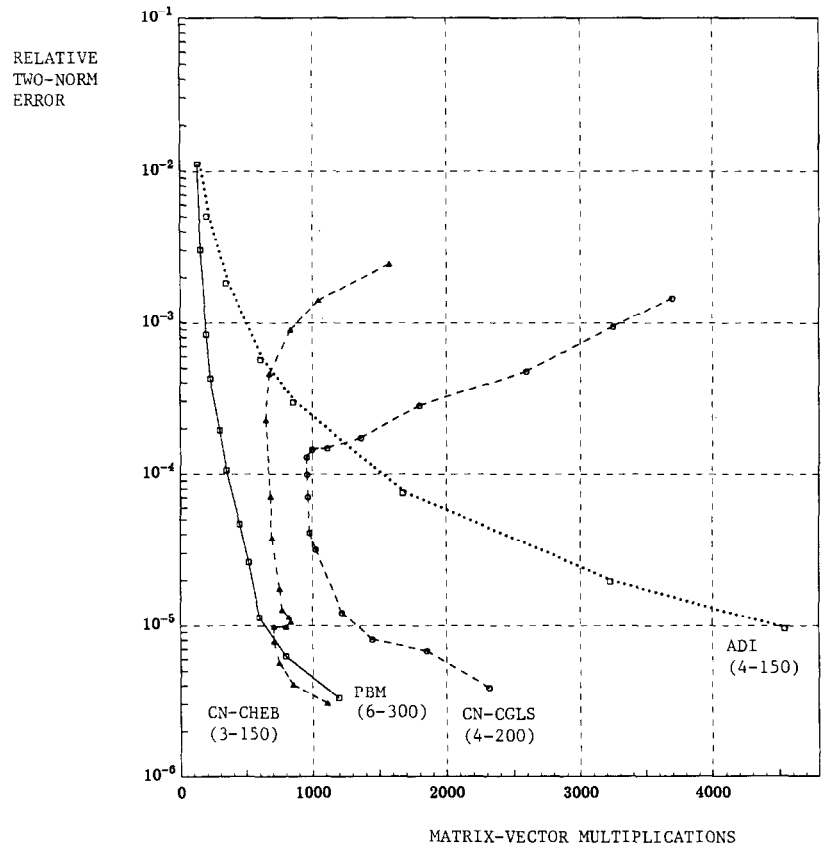

Fig. 2. Accuracy versus computational work for a range of step sizes and several schemes applied to equation (6.1).

conjugate gradients applied to the normal equations. For details on these methods the reader is referred to the respective references cited above.

Figure 2 shows the result of applying the polynomial method, ADI and Crank–Nicolson to equation (6.1) for a range of time steps $T/\Delta t$ which for each method is listed in parentheses below its name. In tracing the curves in a downward direction one reduces the uniform step size Δt and thereby arrives at a better final accuracy. The vertical coordinate gives the relative two-norm error of the solution at time $T = 1.0$, i.e., $(\Sigma_{i,j,k}(U_{ijk} - u(ih, jh, kh))^2/\Sigma_{i,j,k}u(ih, jh, kh)^2)^{1/2}$, where $h = 1/20$ so that the total number of unknowns is $19^3 = 6859$. The horizontal coordinate gives the total number of matrix–vector multiplications required by PBM, CN-CHEB (Crank–Nicolson in conjunction with CHEBYCODE) and CN-CGLS (Crank–Nicolson with CGLS). No matrix–vector multiplications are carried out by the ADI algorithm; for this scheme the CPU-times (obtained on one CPU of a CRAY X-MP/24) form the basis for determining an equivalent number of matrix–vector multiplications by comparison with the times required by PBM.

It is interesting to observe that although a reduction in Δt implies a greater number of time steps, the total amount of work can still be less, as exhibited by CN-CHEB and especially CN-CGLS. This is because the rates of convergence of these iterative methods depend to some

Table 5

Step size Δt	Average number of matrix–vector multiplications		
	PBM	CN-CHEB	CN-CGLS
$\frac{1}{10}$	15.2	65.0	180.2
$\frac{1}{25}$	9.0	27.8	38.2
$\frac{1}{50}$	7.0	16.8	20.5
$\frac{1}{100}$	5.2	8.6	14.5

extent on the condition number of $(I - \frac{1}{2}\Delta t\, L)$ and $(I - \frac{1}{2}\Delta t\, L)^{\mathrm{T}}(I - \frac{1}{2}\Delta t\, L)$, respectively, and in this example the condition of $(I - \frac{1}{2}\Delta t\, L)$ deteriorates rapidly with increasing Δt (see [19]). Of course $\mathrm{cond}_2[(I - \frac{1}{2}\Delta t\, L)^{\mathrm{T}}(I - \frac{1}{2}\Delta t\, L)] = [\mathrm{cond}_2(I - \frac{1}{2}\Delta t\, L)]^2$ which helps explain the relatively poor performance of CN-CGLS. It must also be borne in mind that a single iteration of CGLS requires two matrix–vector multiplications, one involving $(I - \frac{1}{2}\Delta t\, L)$ and the other $(I - \frac{1}{2}\Delta t\, L)^{\mathrm{T}}$, whereas CHEBYCODE requires only multiplication by $(I - \frac{1}{2}\Delta t\, L)$.

The polynomial method also converges more slowly for greater values of Δt; this is natural since the spectrum of $\Delta t\, L$ will then cover a larger domain in $\mathbb{C}$. However, the asymptotic rate of convergence of the iteration is never limited by the spectral properties of $\Delta t\, L$ (see Section 4), and indeed it is seen that in spite of the increase in work per time step the total amount of work still decreases as Δt is increased. Table 5 shows the average number of matrix–vector multiplications per time step for different values of Δt for PBM, CN-CHEB, and CN-CGLS. Table 6 on the other hand shows the accuracies obtained with these values of Δt (in terms of the relative two-norm error at $T = 1.0$) and it is seen that for a given step size Crank–Nicolson is somewhat more accurate than either PBM or ADI. But in terms of the computational work required to achieve a given accuracy the polynomial method is still competitive.

In the remainder of this section we specify the parameter settings for the various schemes (note that ADI does not require any such settings):

– PBM uses the Chebychev polynomials of the second kind and is terminated as soon as $\| \bar{u}_{k+1} - \bar{u}_k \|_2 / \| \bar{u}_{k+1} \|_2 \leqslant 10^{-6}$ for all Δt. As regards the choice of foci (A, B), these are placed at Δt (2.5 − 70i) and Δt (2.5 + 70i), where R (the rectangle representing the spectrum of $\Delta t\, L$) is roughly estimated from the symmetric and skew-symmetric parts of $L(t)$ with $t = 0.5$.

– CGLS is terminated exactly analogous to PBM for all values of Δt.

– CHEBYCODE has a different stopping criterion built into its code which requests the specification of a number *ERBND* such that on return, the two-norm error in the final iterate

Table 6

Step size Δt	Relative two-norm error			
	PBM	CN-CHEB	CN-CGLS	ADI
$\frac{1}{10}$	$0.303 \cdot 10^{-2}$	$0.226 \cdot 10^{-3}$	$0.282 \cdot 10^{-3}$	$0.183 \cdot 10^{-2}$
$\frac{1}{25}$	$0.428 \cdot 10^{-3}$	$0.376 \cdot 10^{-4}$	$0.131 \cdot 10^{-3}$	$0.295 \cdot 10^{-3}$
$\frac{1}{50}$	$0.106 \cdot 10^{-3}$	$0.106 \cdot 10^{-4}$	$0.319 \cdot 10^{-4}$	$0.748 \cdot 10^{-4}$
$\frac{1}{100}$	$0.262 \cdot 10^{-4}$	$0.407 \cdot 10^{-5}$	$0.809 \cdot 10^{-5}$	$0.197 \cdot 10^{-4}$

over that in the initial guess is hopefully less than or equal to *ERBND*. We have set $ERBND = 10^{-6}$ for all values of Δt. The only other parameters required by CHEBIT (the routine we use in CHEBYCODE) are the ellipse parameters D and $C2$ which are updated adaptively by the routine and are saved and passed on between successive calls to CHEBIT. Initially $C2 = 0$ and $D = 1 + 2.5\,\Delta t$ since we estimate $1 + 2.5\,\Delta t$ to lie in the heart of $(I - \frac{1}{2}\,\Delta t\,L)$'s spectrum at $t = 0$.

References

[1] Y. Adam, Highly accurate compact implicit methods and boundary conditions, *J. Comput. Phys.* **24** (1977) 10–22.

[2] S.F. Ashby, CHEBYCODE: A FORTRAN implementation of Manteuffel's adaptive Chebyshev algorithm, Report No. UIUCDCS-R-85-1203, 1985.

[3] J.R. Cash, Two new finite difference schemes for parabolic equations, *SIAM J. Numer. Anal.* **21** (1984) 433–446.

[4] J.C. Cavendish, W.E. Culham and R.S. Varga, A comparison of Crank–Nicolson and Chebychev rational methods for numerically solving linear parabolic equations, *J. Comput. Phys.* **10** (1972) 354–368.

[5] M. Ciment, S.H. Leventhal and B.C. Weinberg, The operator compact implicit method for parabolic equations, *J. Comput. Phys.* **28** (1978) 135–166.

[6] P.J. Davis, *Interpolation and Approximation* (Blaisdell, New York, 1963) and (Dover, New York, 1975).

[7] L. Fox, *Numerical Solution of Ordinary and Partial Differential Equations* (Pergamon, Oxford, 1962).

[8] G.H. Golub and C.F. Van Loan, *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore, MD, 1983).

[9] A.R. Gourlay and S. McKee, The construction of hopscotch methods for parabolic and elliptic equations in two space dimensions with a mixed derivative, *J. Comput. Appl. Math.* **3** (1977) 201–206.

[10] A.R. Gourlay and J.Ll. Morris, The extrapolation of first order methods for parabolic partial differential equations, II, *SIAM J. Numer. Anal.* **17** (1980) 641–655.

[11] A.S. Householder, *The Theory of Matrices in Numerical Analysis* (Blaisdell, New York, 1964) and (Dover, New York, 1975).

[12] D.P. Laurie, *Numerical Solution of Partial Differential Equations: Theory, Tools and Case Studies* (Birkhäuser, Basel, 1983).

[13] J.D. Lawson and J.Ll. Morris, The extrapolation of first order methods for parabolic partial differential equations, I, *SIAM J. Numer. Anal.* **15** (1978) 1212–1224.

[14] R.J. LeVeque, Intermediate boundary conditions for LOD, ADI and approximate factorization methods, NASA Contractor Report 172591 (ICASE Report No. 85-21), 1985.

[15] A.R. Mitchell and D.F. Griffiths, *The Finite Difference Method in Partial Differential Equations* (Wiley, Chichester, 1980).

[16] C.B. Moler and C.F. Van Loan, Nineteen dubious ways to compute the exponential of a matrix, *SIAM Rev.* **20** (1978) 801–836.

[17] C.C. Paige and M.A. Saunders, LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Software* **8** (1982) 43–71.

[18] Y. Saad, Least squares polynomials in the complex plane with applications to solving sparse nonsymmetric matrix problems, Research Report YALEU/DCS/RR-276, 1983.

[19] M.J. Schaefer, A polynomial based iterative method for linear parabolic equations, Ph.D. thesis, Dept. of Computer Science, University of Illinois at Urbana-Champaign, 1987.

[20] G. Szegö, *Orthogonal Polynomials* (Amer. Mathematical Soc., Providence, RI, 4th ed., 1975).

[21] R.S. Varga, *Matrix Iterative Analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1962).